



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/667,816	09/22/2003	John Phenix	14846-21	2173
7590	01/31/2008		EXAMINER RADTKE, MARK A	
Michael B. Johannesen, Esq. Lowenstein Sandler, P.C. 65 Livingston Avenue Roseland, NJ 07068			ART UNIT 2165	PAPER NUMBER
			MAIL DATE 01/31/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/667,816
Filing Date: September 22, 2003
Appellant(s): PHENIX, JOHN

MAILED

JAN 31 2007

Technology Center 2100

Raymond G. Cappo
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 10 September 2007 appealing from
the Office action mailed 8 March 2007

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,826,716	Mason	11-2004
6,662,312	Keller et al.	12-2003

"Java Performance Tuning", Section 7.5, "Recursion and Stacks", Jack Sharazi, published September 2000, available online at:
<http://proquest.safaribooksonline.com/0596000154>

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1, 3-5 and 7-9 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are rejected as falling under the judicial exception of an abstract idea which lacks a useful, concrete and tangible result. A claimed series of steps or acts that do not result in a useful, concrete

and tangible result are not statutory within the meaning of 35 U.S.C. 101. In the instant case, the claims recite "determining". However, no useful, concrete and tangible result is disclosed in the instant application as originally filed. Appellants have not shown how or why one of ordinary skill in the art would be aware of any undisclosed useful, concrete and tangible results (such as the "null output" described in Appellant's brief).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1, 3-5 and 7-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mason (U.S. Pat. No. 6,826,716) in view of Shirazi ("Java Performance Tuning", Section 7.5, "Recursion and Stacks", published September 2000, available online at: <http://proquest.safaribooksonline.com/0596000154>) and further in view of Keller et al. (U.S. Pat. No. 6,662,312).

As to claim 1, Mason teaches a computer implemented method for comparing a first object and a second object in an object-oriented operating system (see Abstract and Summary of the Invention) comprising the steps of:

(a) determining whether the first object (see figure 4, element 412, "EJB Module DD") is equal to the second object (see figure 4, element 410, "QOS") (See column 11, lines 28-34. Elements 414 and 406, which are sub-elements of elements 412 and 410, respectively, are compared by testing for equality.) and if said objects are not equal (See column 11, lines 38-43. If the test of step (a) succeeds, the program proceeds to test if elements 430 and 404 are equal.);

(b) obtaining one or more methods from said first object and said second object (see column 11, lines 38-43. The methods are obtained via reflection. See column 3, lines 48-54.);

(c) determining whether the one or more methods from said first object are equal to the one or more methods from said second object (see column 11, lines 38-43); and

(d) performing steps (b) and (c) until all of the methods for the first object and the second object have been obtained (See column 8, line 47, "For Each method-permission element". Each method described in element 410 will be tested. All methods for element 404 will be obtained during the reflection step.)

(e) generating a document comprising a listing of differences between the methods (See column 12, line 45, "reporting (444)" and see column 15, lines 18-20); and

(g) displaying the human-readable form to a user (all computers have displays).

Mason does not explicitly teach recursively performing steps.

Shirazi teaches recursively performing steps (see pages 1-2).

Therefore, it would have been obvious to one of ordinary skill in the relevant art at the time the invention was made to have modified Mason by the teaching of Shirazi because "the iterative method actually takes 5% longer than the recursive method" (see Shirazi, page 2, paragraph 2, lines 1-2).

The combination of Mason and Shirazi as discussed supra still does not explicitly teach (f) transforming the document into human-readable form.

Keller et al. teaches (f) transforming the document into human-readable form (see figure 4).

It would have been obvious to one of ordinary skill in the relevant art at the time the invention was made to have modified the combination of Mason and Shirazi as discussed supra by the teaching of Keller et al. because being human-readable is a desirable feature of reports because reports are meant to be read by a computer operator and the computer operator could not read them unless they were in a human-readable form (see Mason, column 12, line 45, "reporting").

As to claim 3, Mason, as modified, teaches wherein step (b) further comprises storing names of said methods (see column 13, lines 64-65, "VAR METHODNAME"), step (c) comprises storing indicia representing the determination whether the methods are the same (see column 14, line 7, "For each match found") and step (e) comprises generating a document from said stored names and indicia (see column 15, line 21, "e1.printStackTrace()").

As to claim 4, Mason, as modified, teaches wherein said object-oriented operating system comprises JAVA (see Abstract) and step (a) comprises running an equality method on said objects (See Examiner's comments regarding claim 1. Testing whether objects are equal implies running an equality method).

As to claim 5, Mason, as modified, teaches wherein said object-oriented operating system comprises JAVA (see Abstract) and step (c) comprises running an equality method on said objects (See Examiner's comments regarding claim 1. Testing whether methods are equal implies running an equality method).

As to claim 7, Mason, as modified, teaches human-readable form (see Examiner's comments regarding claim 1).

Mason, as modified, does not expressly teach comprising XML. However, these differences are only found in the nonfunctional descriptive material and are not functionally involved in the steps recited. The determination of human-readable would be performed the same regardless of file format. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, (see *In re Gulack*, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); *In re Lowry*, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994)).

Therefore, it would have been obvious to a person of ordinary skill in the relevant art at the time the invention was made to display information to a human operator based on any type of file format, because such data does not functionally

relate to the steps in the method claimed and because the subjective interpretation of data does not patentably distinguish the claimed invention.

As to claim 8, Mason, as modified, teaches wherein transforming the document into human-readable form comprises transforming the document into a web page (see column 6, lines 62-67 and see column 10, lines 12-14).

As to claim 9, Mason, as modified, teaches wherein said object-oriented operating system comprises JAVA (see Abstract) and step (b) comprises invoking a get...() method of each object (JavaBeans consist of publicly-exposed set and get methods).

(10) Response to Argument

In response to Appellant's arguments that claims 1, 3-5 and 7-9 are statutory subject matter under 35 U.S.C. 101, the arguments have been fully considered but are not deemed persuasive.

Appellant argues that it is known in the art that many commands on the Unix operating system return a null result to return information to the user. Although Examiner does not disagree with Appellant's factual statements regarding the operation of the Unix operating system, Examiner does disagree with Appellant's assertion that a null result is useful, concrete and tangible. Appellant's argument hinges on an assertion that the lack of a result constitutes a result. The American Heritage College Dictionary

defines null as "Of no consequence, **effect**, or value" (emphasis added). This corresponds to a null result in the Unix operating system because a null result produces no consequence or effect. Nothing is displayed to the user and (in the case of the diff command and the claimed invention) no files or memory locations are modified. The state of the system is unchanged. An unaltered system that produces no output is not useful, concrete or tangible.

In the case when the objects under comparison are equal, the claimed invention will merely "determine" that they are equal. No useful, concrete or tangible result is produced by the method.

In response to Appellant's arguments that "*Mason* describes a system for testing various parameters of Java Web applications to determine if the applications succeed" (emphasis in original) and thus "*Mason* is very different from the method claimed" (emphasis in original), the arguments have been fully considered but are not deemed persuasive.

Appellant is arguing that Mason and the claimed invention differ in intended use. First, it is noted that the preamble of the claim is usually not afforded patentable weight, so Appellant's argument is moot. Regardless, the claimed invention and the invention of Mason are in fact from similar fields of use. In the invention of Mason, a Java object (first object) and a description of a test for that object (second object) are specified by a user. The structure and behavior of the created object is compared to the expected structure and behavior described in the test. Both the invention of Mason and the

claimed invention operate on the Java operating system. They both compare two objects. Mason compares an instance of an object with a description of the expected object. The **disclosed** invention compares two objects with the expectation that there are differences between the objects. The invention of Mason compares two objects with the expectation that there are not differences between the two. However, both inventions deal with the same problem (comparing two objects) and arrive at the same solution (besides obvious differences taught by Keller and Shirazi).

In response to Appellant's argument that Mason does not teach "determining whether the first object is equal to the second object; and if said objects are not equal", the arguments have been fully considered but are not deemed persuasive.

As described at line 33 of column 11, Mason determines if the name specified in enterprise-bean 414 is equal to the name specified in ejb-name 406. If this operation fails, then the algorithm of Mason would terminate because the comparisons described in the rest of the column 11 could not be performed. As described at lines 34-38 of Mason, elements 406 and 414 have sub-elements (i.e., children) that are compared later. Before children could be compared, it must be confirmed that the parents are equal. If the parents are unequal, then the inventions are operating on two different classes of objects, so a comparison of methods (that comprise objects) would be irrelevant.

In response to Appellant's argument that Mason does not teach "obtaining one or more methods from said first object and said second object", the arguments have been fully considered but are not deemed persuasive.

Appellant states that the cited portion of Mason is irrelevant to the limitation at hand. The rejection has been further clarified in the prior art rejection above. At lines 40-46 of column 12, Mason describes invoking methods on the object under test. A method that is able to be "invoked" has been "obtained". The word "obtain" has no particular meaning in the art of computer programming beyond its ordinary use. One of ordinary skill in the art would reasonably determine methods to be "obtained" if they exist anywhere in the system. Numerous mentions are made throughout columns 11 and 12 to operations performed on the methods of elements 406 and 414. Thus, they are obtained.

In response to Appellant's argument that Mason does not teach "determining whether the one or more methods from said first object are equal to the one or more methods from said second object", the arguments have been fully considered but are not deemed persuasive.

At lines 42 of column 11, Mason describes obtaining the name of a method that comprises the first object through the use of reflection. This method name is a child of enterprise-bean 412 and is compared to method-name 404, which is a child of ejb-name 406. "Such embodiments typically include identifying [...] a method signature (430)

having a method name equal to the first <method-name> sub-element (404) value" (emphasis added).

In response to Appellant's argument that Mason does not teach "recursively performing steps (b) and (c) until all of the methods for the first object and the second object have been obtained", the arguments have been fully considered but are not deemed persuasive.

See column 8, line 47: "For Each method-permission element". Each method described in element 410 will be tested. All methods for element 404 will be obtained during the reflection step. Appellant goes on to argue that this code does not perform the steps recursively. This is admitted by Examiner in the prior art rejection. Shirazi is cited for teaching this capability. Appellant further argues "[t]here is only one object involved in *Mason*" (emphasis in original). There are many objects in Mason including an EJB Module deployment descriptor 404 and the QOS elements 410. These objects describe other objects such as JavaBeans.

In response to Appellant's argument that Mason does not teach "generating a document comprising a listing of differences between the methods", the arguments have been fully considered but are not deemed persuasive.

This difference list is generated at lines 18-20 of column 15 in Mason. Each time a method invocation fails, an Exception will be thrown. The "catch" block then outputs a message to the user saying "UnExpected exception during EJB method invocation:"

and a stack trace. The stack trace can be interpreted by one of ordinary skill in the art to determine where errors occurred. In the case of the invention of Mason, errors will be displayed for each mismatched method. See the "Display the Stack Trace" reference, cited by Examiner on 8 March 2007.

In response to Appellant's argument that Shirazi "does not make up for the shortcomings of *Mason*" (emphasis in original), the arguments have been fully considered but are not deemed persuasive.

Shirazi is cited solely for teaching that the invention of Mason can be programmed using a recursive algorithm. Appellant has not presented any argument regarding this aspect of Shirazi.

In response to Appellant's argument that Keller "fails to correct the deficiencies of either *Mason* or *Shirazi*" (emphasis in original), the arguments have been fully considered but are not deemed persuasive.

Appellant has not presented any arguments related to the limitations that Keller was cited for teaching.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Mark A. Radtke/

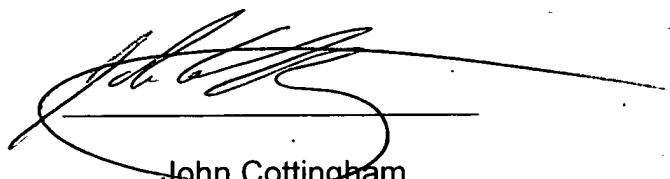
Mark A. Radtke

Appeal Conference held on 3 December 2007. Agreement was reached to proceed to the Board of Appeals and Interferences.

Conferees:



Christian Chace



John Cottingham